

FHWS

# iOS Programmierung (mit Swift)

Peter Braun, Florian Bachmann & Andreas Wittmann  
[@pe\\_braun](https://twitter.com/pe_braun)   [@florianbachmann](https://twitter.com/florianbachmann)   [@anwittmann](https://twitter.com/anwittmann)

Deutsche Telekom AG  
FHWS - Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt  
#FHWSSwift

# Agenda

1. **Introduction** – Organisatorisches
2. **First iOS-Project** – Hello World, **First iOS-Project** – Still Hello World (now with Code 😊)
3. **Swift**, Wait!, What about Objective-C?, Why Swift?
4. **A (not so) Quick Tour**
5. **Documentation**
6. **The basics** – iOS Architecture & more
7. **User Interfaces** – View Controller, Auto Layout & Size Classes
8. **Storyboard & Segues**
9. **Tables & NavigationController**
10. **TabBarController**
11. **Notifications**
12. **PickerViews**
13. **Touches, Gestures, 3D Touch, Peek & Pop**
14. **ScrollView & StackViews**
15. **Networking** – JSON & Dependency Managers
16. **WebKit**
17. **Maps**
18. **Storage & Data persistency** – NSUserDefaults, NSKeyedArchiver & Core Data
19. **ObjC**

# Swift

# Alexander von Below



(ObjC-Swift-Evangelist)

## “Swift ist eine seelenlose Hipster-Sprache”

<https://twitter.com/avbelow>

# Objective-C without the C.

“The language is called Swift  
and it totally rules.”



# Swift. A new language that lets everyone build amazing apps.

Swift is an innovative new programming language for Cocoa and Cocoa Touch. Writing code is interactive and fun, the syntax is concise yet expressive, and apps run lightning-fast. Swift is ready for your next iOS and OS X project — or for addition into your current app — because Swift code works side-by-side with Objective-C.

# About Swift (1/2)

‘Swift is friendly to new programmers. It is the first industrial-quality systems programming language that is as expressive and enjoyable as a scripting language. It supports playgrounds, an innovative feature that allows programmers to experiment with Swift code and see the results immediately, without the overhead of building and running an app.’

Balloons — Balloons.playground — Edited

```

func didMoveToView(scene : SKScene,
                  delegate : SKPhysicsContactDelegate) {

    // ===== Blimp Control =====
    yOffsetForTime = { i in
        return 80 * sin(i / 10.0)
    }

    // ===== Scene Configuration =====
    // Set up balloon lighting and per-pixel collisions.
    balloonConfigurator = { b in
        b.physicsBody.categoryBitMask = CONTACT_CATEGORY
        b.physicsBody.fieldBitMask = WIND_FIELD_CATEGORY
        b.lightingBitMask = BALLOON_LIGHTING_CATEGORY
    }

    // Load images for balloon explosion.
    balloonPop = (1...4).map {
        SKTexture(imageNamed: "explode_0\($0)")
    }

    // Install turbulent field forces.
    var turbulence = SKFieldNode.noiseFieldWithSmoothness(0.7,
                                                          animationSpeed:0.8)
    turbulence.categoryBitMask = WIND_FIELD_CATEGORY
    turbulence.strength = 0.21
    scene.addChild(turbulence)

    cannonStrength = 210.0

    // ===== Scene Initialization =====
    // Do the rest of the setup and start the scene.
    setupHero(scene, delegate)
    setupFan(scene, delegate)
    setupCannons(scene, delegate)
}

func handleContact(bodyA : SKSpriteNode,
                  bodyB : SKSpriteNode)| {

    if (bodyA == hero) {
        bodyB.normalTexture = nil
        bodyB.runAction(removeBalloonAction)
    } else if (bodyB == hero) {
        bodyA.normalTexture = nil
        bodyA.runAction(removeBalloonAction)
    }
}

```

(Function)  
(1058 times)

(Function)  
(55 times)

[SKTexture, SKTexture, SKTe...  
(4 times)

SKNoiseFieldNode  
SKNoiseFieldNode  
SKNoiseFieldNode  
(GameScene ((Function)) {(F...  
210.0

let y = 80 \* sin(x)

Chris Lattner sagt, Playground wurde inspiriert durch die Talks von Bret Victor:  
<http://worrydream.com/>  
<https://www.youtube.com/watch?v=8pTEmbeENF4>

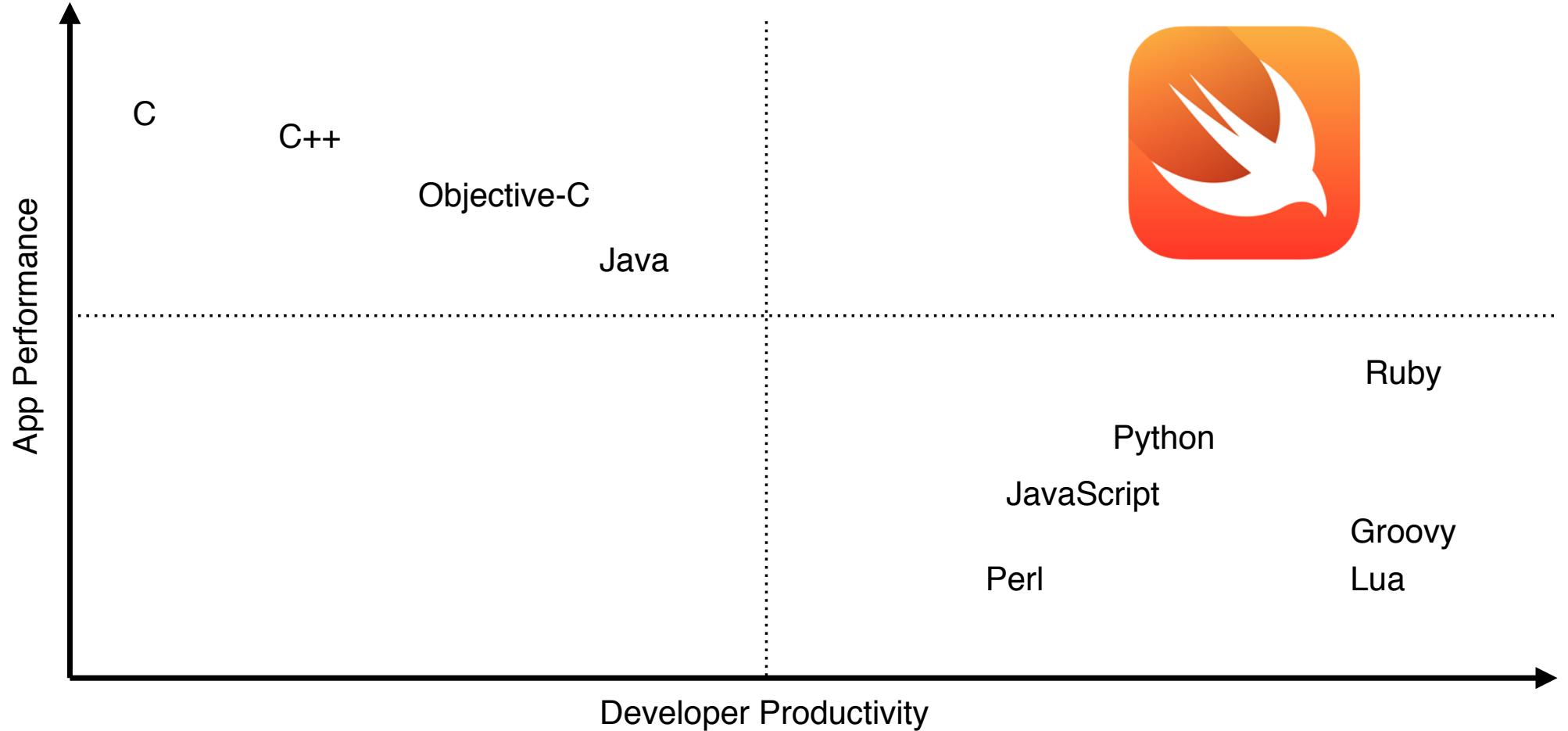
# About Swift (2/2)

‘Swift combines the best in modern language thinking with wisdom from the wider Apple engineering culture. The compiler is optimized for performance, and the language is optimized for development, without compromising on either. It’s designed to scale from “hello, world” to an entire operating system. All this makes Swift a sound future investment for developers and for Apple.’

# Swift

Paradigm	Multi-paradigm (Protocol-oriented programming, object-oriented, functional, imperative, block structured)
Designed by	Chris Lattner and Apple Inc.
Developer	Apple Inc.
First appeared	June 2, 2014
Stable release	2.1 / October 21, 2015
Typing discipline	Static, strong, inferred
OS	iOS, OS X, watchOS, tvOS, Linux later in 2015
License	Proprietary, transitioning to open source in 2015
Filename extensions	.swift
Website	<a href="http://developer.apple.com/swift/">developer.apple.com/swift/</a>
Influenced by	Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, D
Influenced	Rust

Source: [Wikipedia - Swift](#)



# Swift a modern Objective-C

Swift has many other features to make your code more expressive:

- Closures unified with function pointers
- Tuples and multiple return values
- Generics
- Fast and concise iteration over a range or collection
- Structs that support methods, extensions, protocols.
- Functional programming patterns, e.g.: map and filter



**Set up trivial parser support stuff and sketch out a few ...**

lattner committed on Jul 18, 2010



[5450892](#)



**stub out a parser hook. ...**

lattner committed on Jul 18, 2010



[05f5028](#)



**add numeric constant support, we can now lex exciting stuff ...**

lattner committed on Jul 18, 2010



[3cd9e46](#)



**implement lexer support for identifiers and our 2 keywords so far. ...**

lattner committed on Jul 18, 2010



[6236dea](#)



**add some more punctuator characters and // comments. ...**

lattner committed on Jul 18, 2010



[80ba9c7](#)



**start lexing trivial tokens, discard whitespace between them. ...**

lattner committed on Jul 18, 2010



[461f41e](#)



**Start setting up the lexer object. ...**

lattner committed on Jul 18, 2010



[db8e130](#)



**consolidate lexer stuff into the Parser lib. ...**

lattner committed on Jul 18, 2010



[874badb](#)



**sketch out lexer and token interfaces. ...**

lattner committed on Jul 18, 2010



[5e88a21](#)



**initial checkin, nothing much to see here. ...**

lattner committed on Jul 18, 2010

18



[afc81c1](#)



**initial swift test ...**

lattner committed on Jul 18, 2010

185



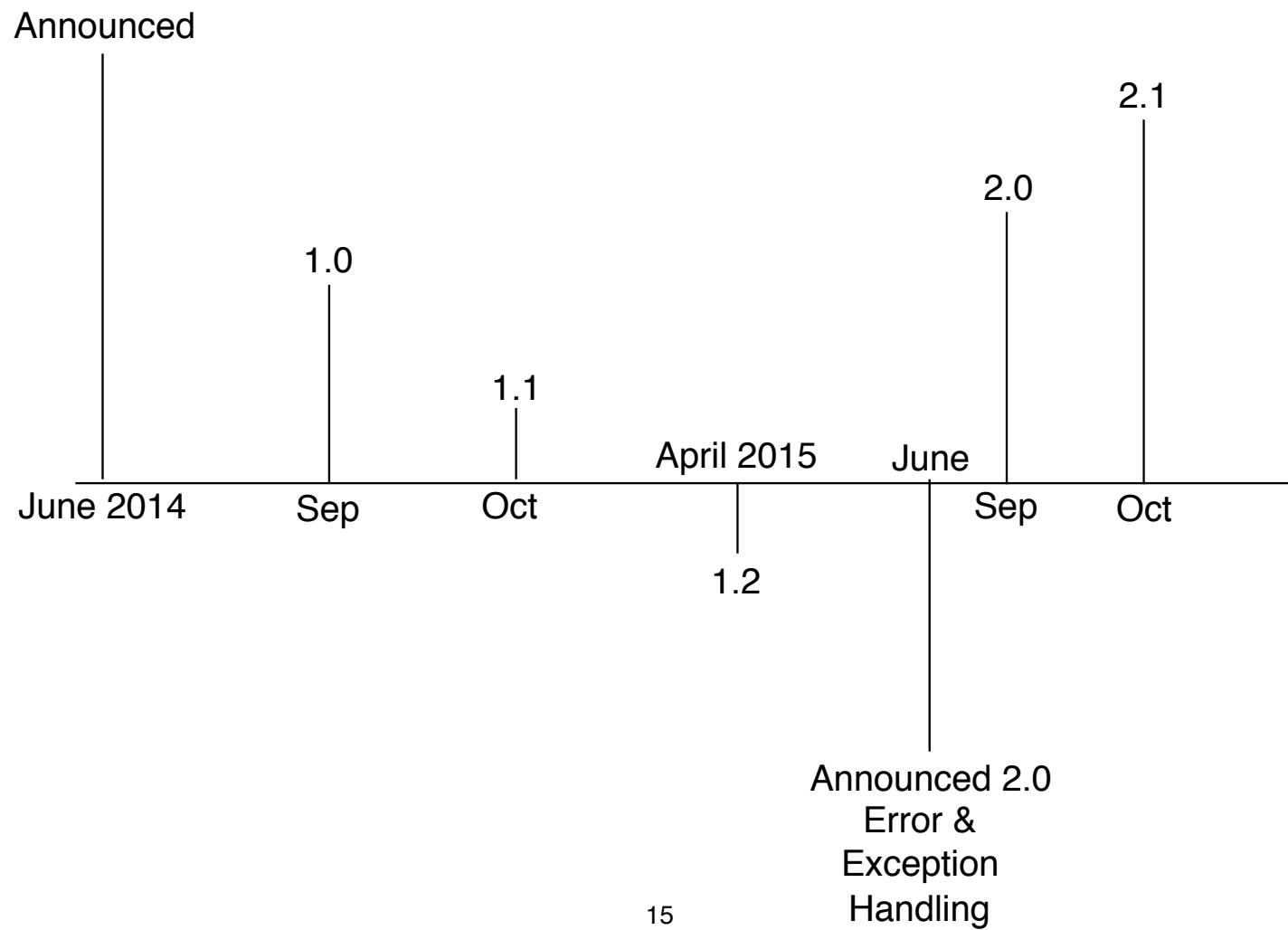
[18844bc](#)



Functional Patterns	Protocols and extensions on structs	Pattern matching
Clear mutability syntax	Read-Eval-Print-Loop (REPL)	Multiple return types
Operator overloading		Int overflow checking
Concise syntax		Object orientation
Native collections		Interactive playground
Trailing closures		Fast iteration
Generics	Playgrounds	Pattern matching
Namespaces	Type inference	Compile to native code
User defined operators	Tuples	Optional types
Swift-specific optimizer	Array bounds checking	Closures



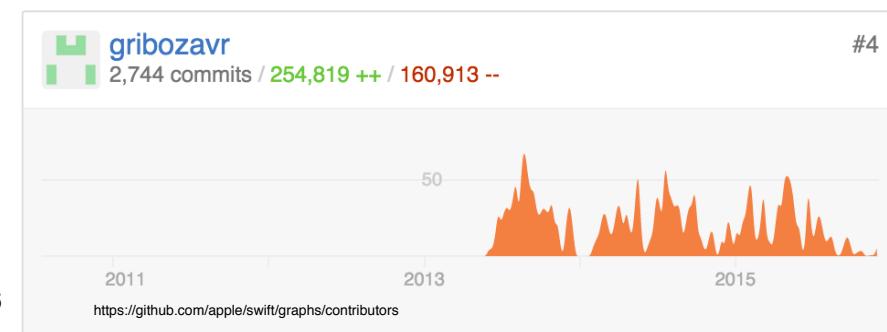
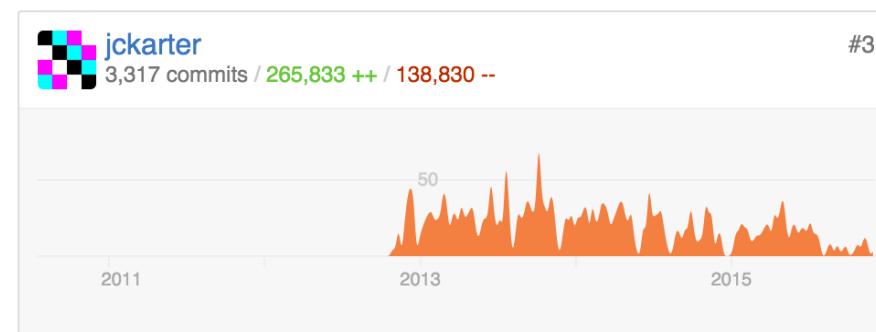
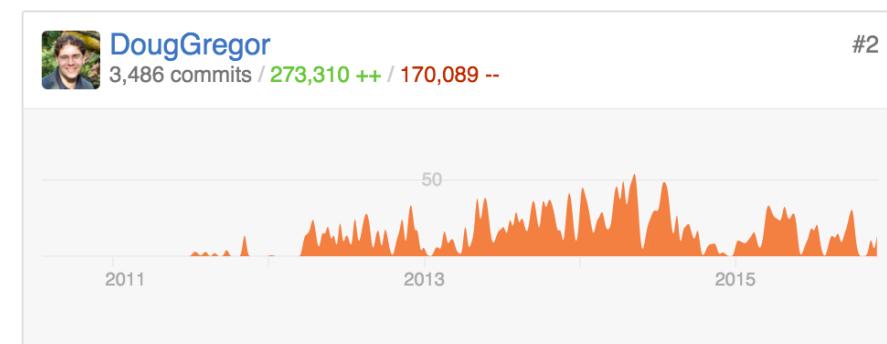
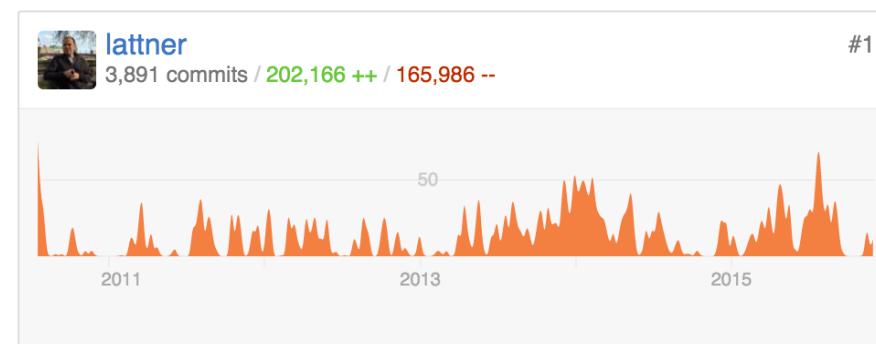
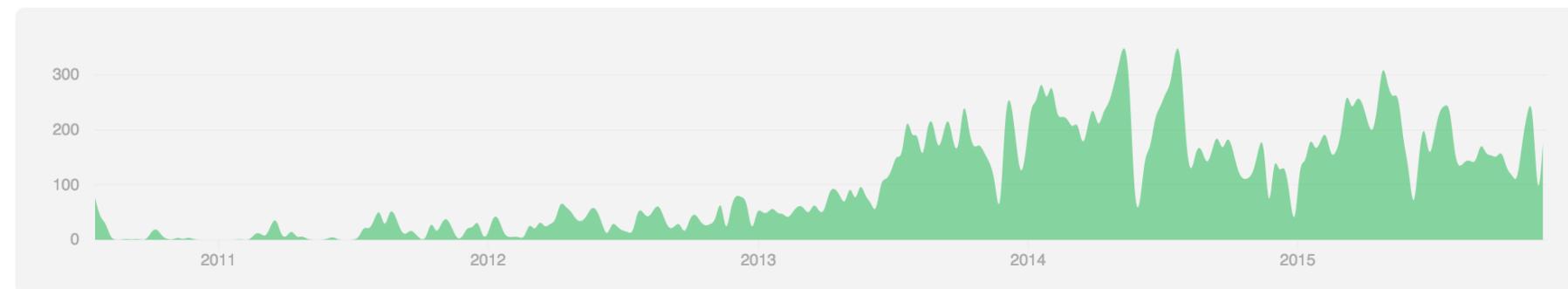
# Swift is evolving fast



Jul 18, 2010 – Dec 4, 2015

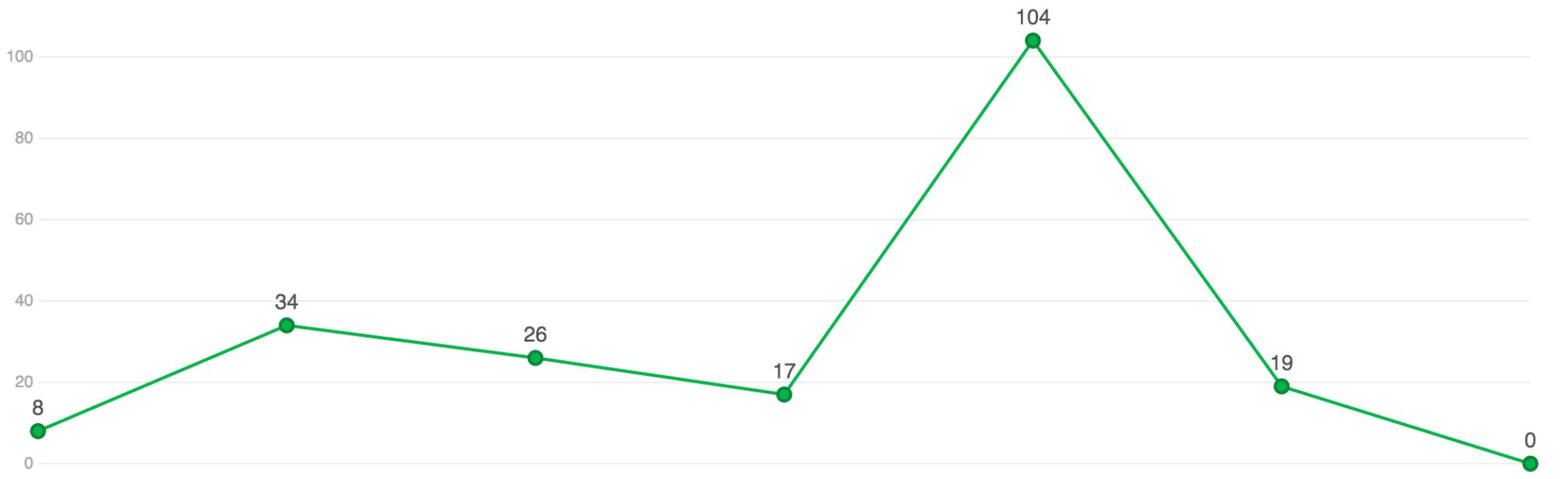
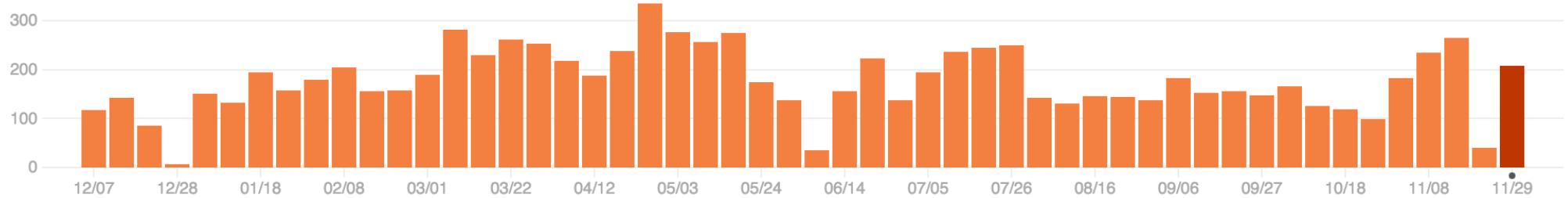
Contributions to master, excluding merge commits

Contributions: **Commits** ▾



Contributors	Commits	Code frequency	Punch card	Network	Members
--------------	---------	----------------	------------	---------	---------

Use ← and → to navigate



# Swift 2.1

Synthesized “header” in Xcode

Swift 1.2 to Migrator

@testable

do {}

Markdown in comments

Faster debug mode

Faster runtime

Error Handling Model

C function pointer APIs

Availability checking

Short compile time



ObjC generics

Mutability warnings

guard

SIMD support

Multi-payload enums

New playgrounds

Improved options sets

Parallelized WMO

Protocol extensions

defer

Nullability in ObjC

Recursive nested functions

Pattern matching in “if”

# Swift 1/3

Modern

Clean syntax

No headers

No semicolons

Multiple return values

Optional arguments

Closures

Generics

# Swift 2/3

# Safe

No uninitialized data

Array bounds checks

Integer Overflow checks

Braces on all control statements

Raw pointers marked “unsafe”

No implicit fall through

# Swift 3/3

## Fast

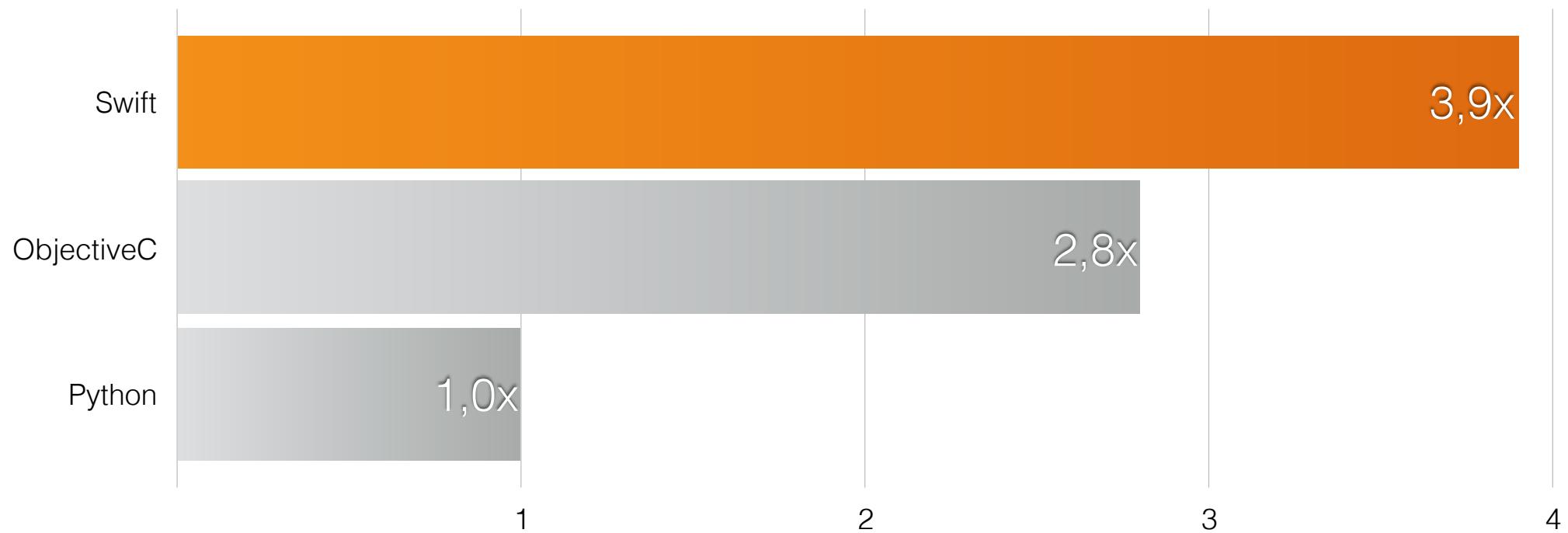
ARM and x86-64 native code

Tuned native collections

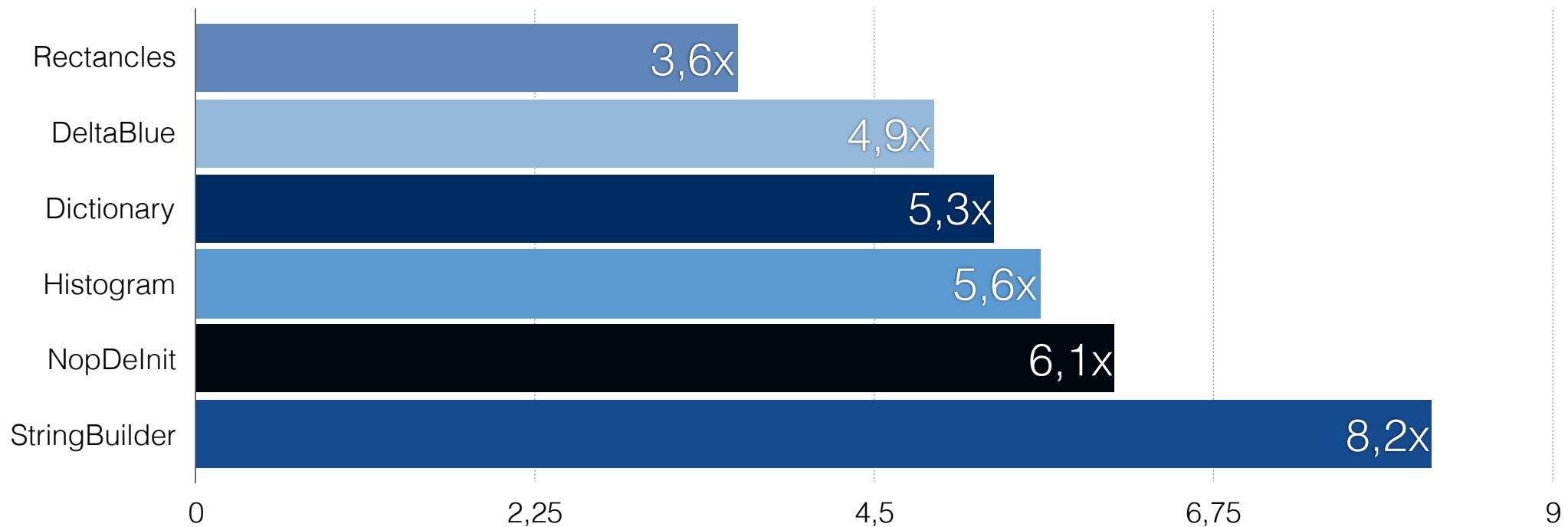
Swift-specific optimizer

C-like procedural performance

# Complex Object Sort\*



# Swift 2.0 Performance Improvement

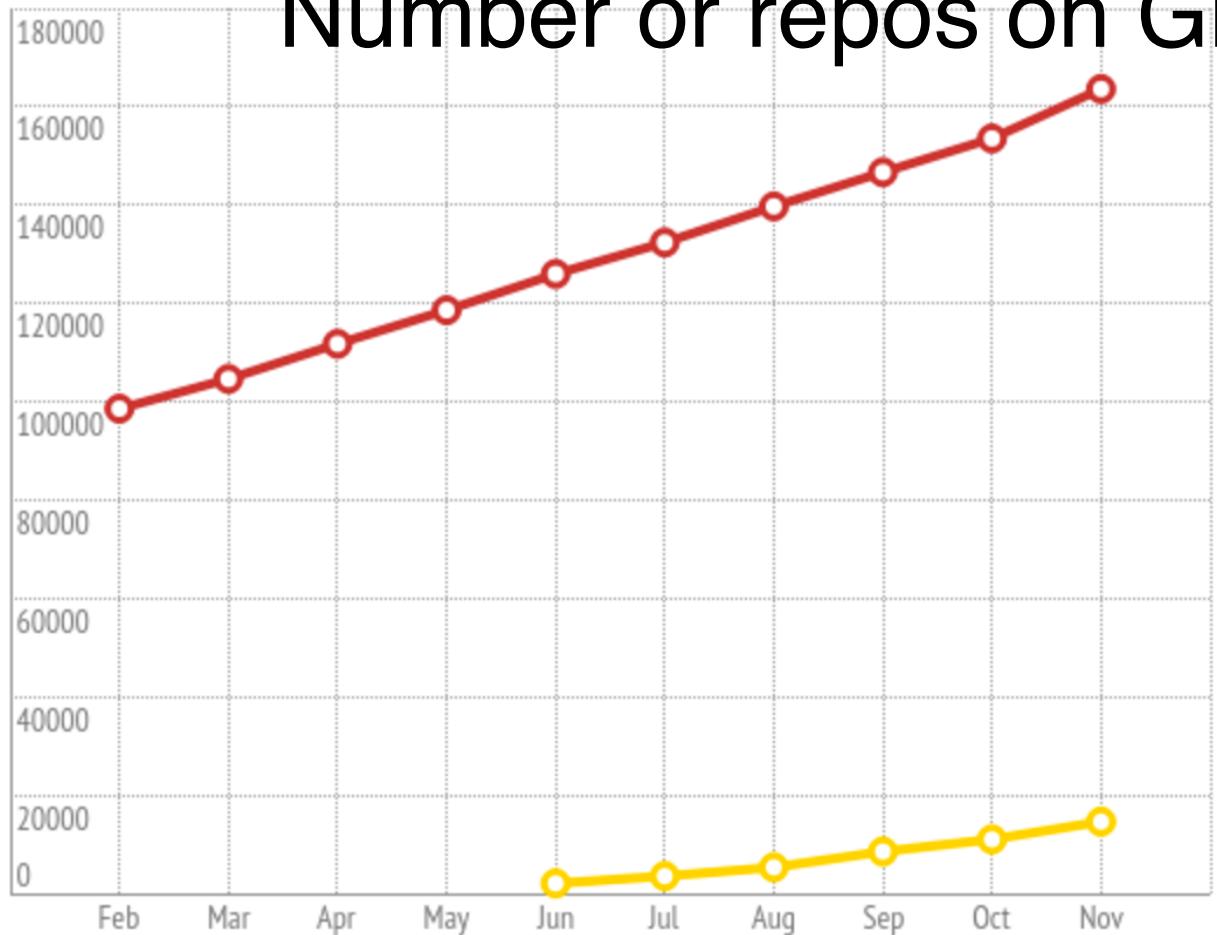


Chris Parker  
in WWDC Session 416:  
"Swift has had 200 native  
speakers for a year".

Wait!  
What about Objective-C?  
Why Swift?

# Swift - Facts 1/7

## Number of repos on Github in 2014



★ 15,000

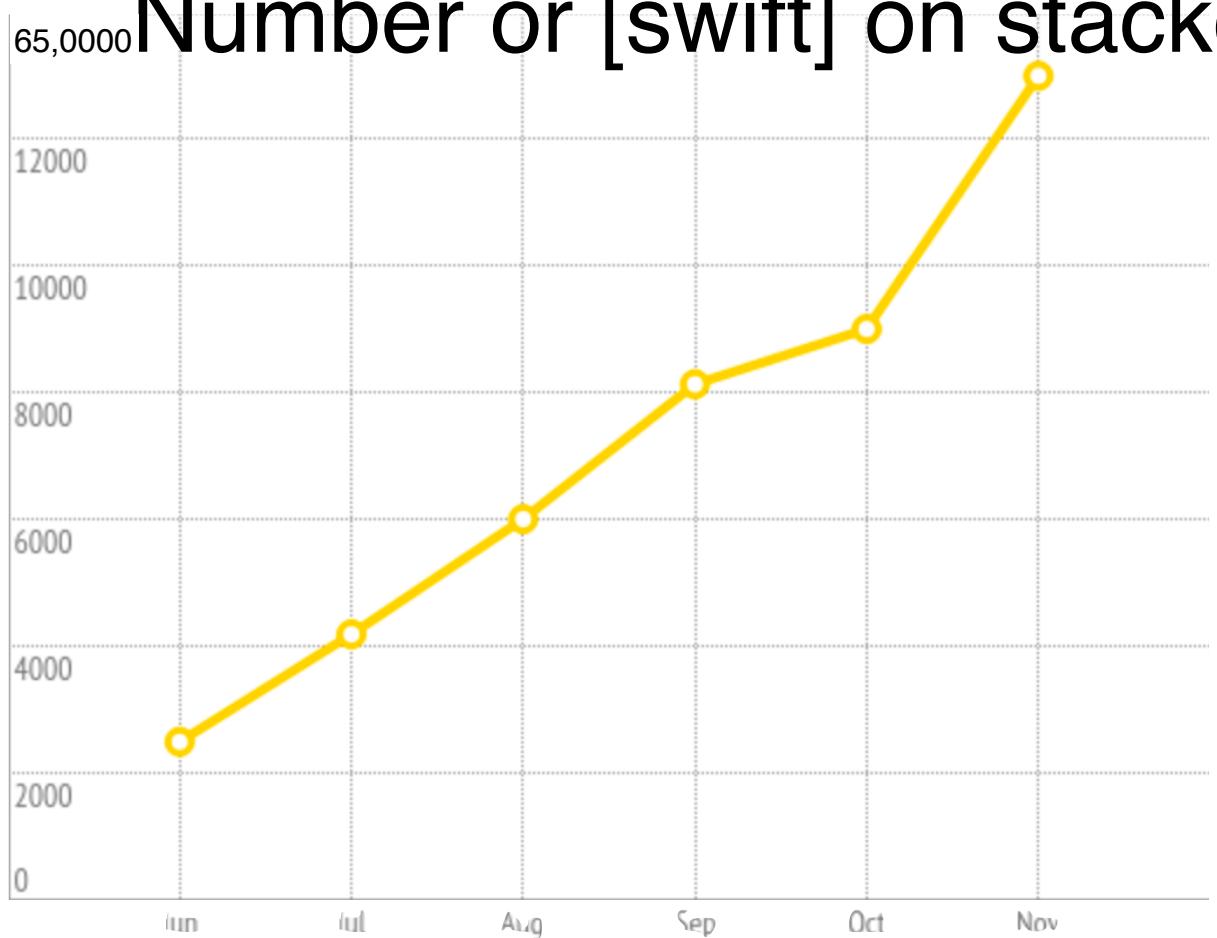
repositories on GitHub

▲ 64

repositories created daily (for  
Objective-C it is 231)

## Swift - Facts 2/7

Number of [swift] on stackoverflow in 2015



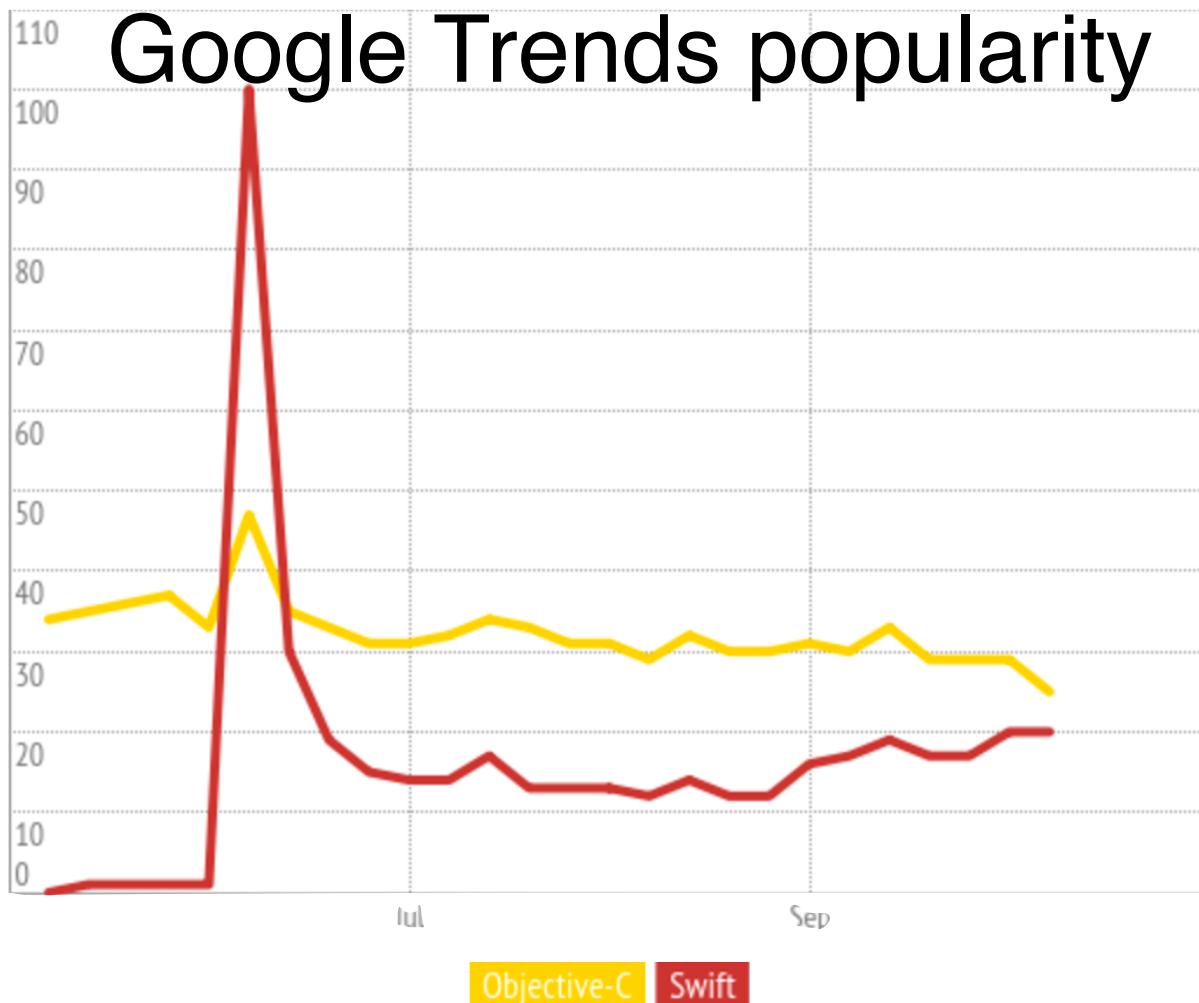
★ 14,000 65,000

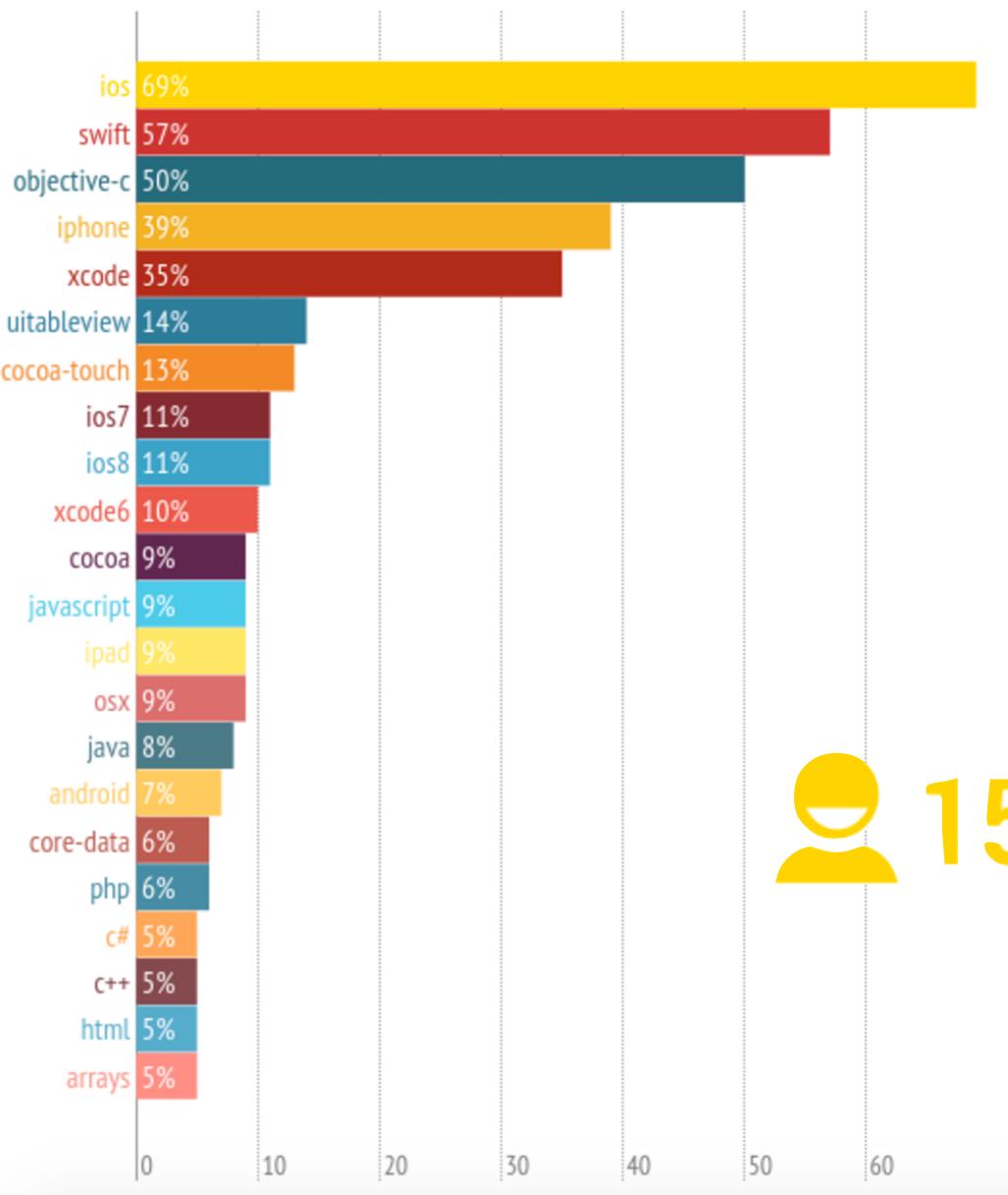
[swift] questions on Stackoverfl

▲ 53

Questions with [swift] tag created  
(for [objective-c] it is 148)

## Swift - Facts 3/7





## Swift - Facts 4/7

### Popular stackoverflow tags in TOP 10 skills of user discussing about [swift]

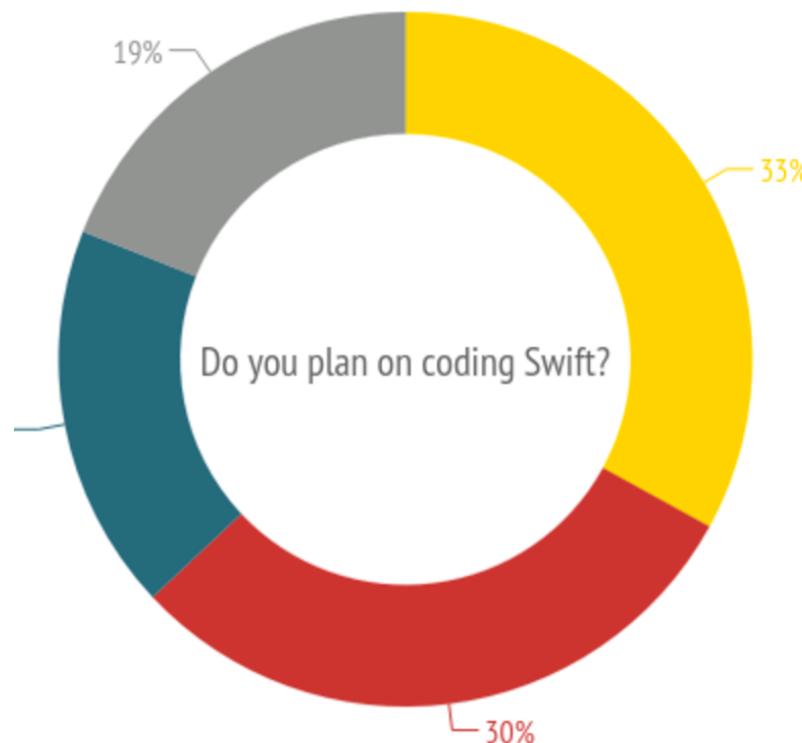


Stackoverflow users who have asked/answered questions about [swift] are not Objective-C developers

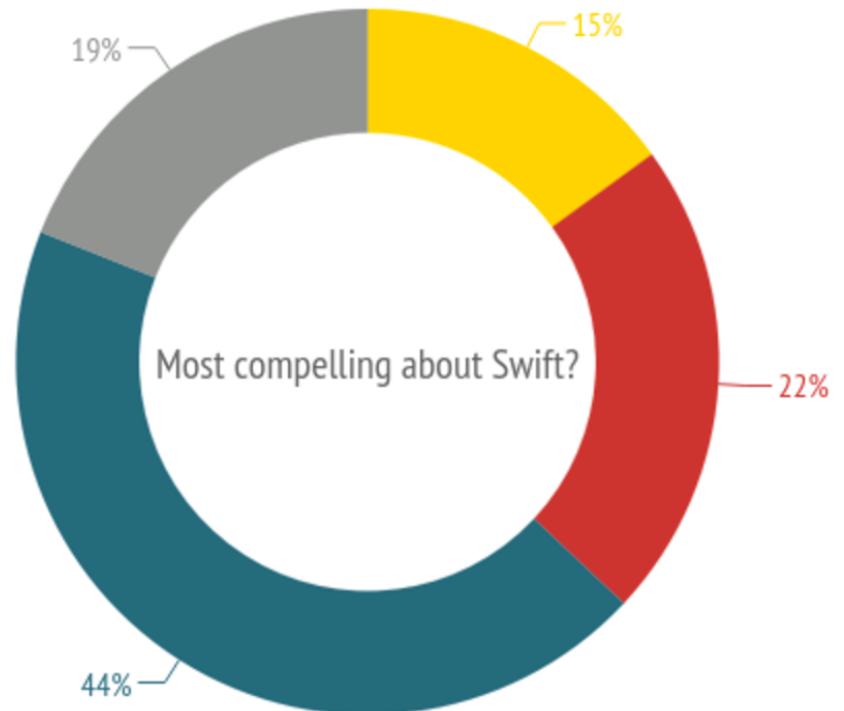
Source: Swift: Interesting Facts by Jetbrains. December, 2014

# Swift - Facts 5/7

## Slashdot survey results (July, 2014)



Yes, as soon as possible | Yes, but it is not a priority  
I won't be coding with Swift | I will delay learning it



Backed by Cocoa/Cocoa Touch | The speed versus Objective-C  
Simple and modern syntax | The preview playground feature

Source: Swift: Interesting Facts by Jetbrains. December, 2014

# Swift - Facts 6/7

## more interesting facts...

▲ 15

O'Reilly books about Swift

110,000

Views of Swift page on Wikipedia for  
the first week after announcement  
(June 2, 20124)

1,000

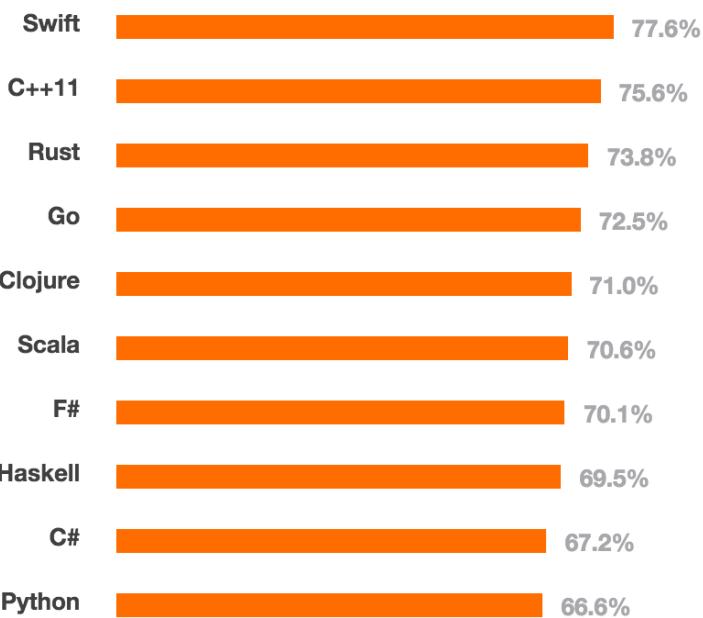
Average number of views of Swift  
page on Wikipedia daily

# Swift - Facts 7/7

Most Loved

Most Dreaded

Most Wanted



*% of devs who are developing with the language or tech that have expressed interest in continuing to develop with it.*

# Seriously, why Swift? 1/6

- OpenSSL Goto error

```
if ((err = SSLHashSHA1.update(&hashCtx,&signedParams))!=0)  
goto fail;  
goto fail;
```

- Objective-C

```
if i < 9  
    print("Hello")
```

- Swift

```
if i < 9 {  
    print("Hello")  
}
```

[http://nearthespeedoflight.com/article/2014\\_09\\_28\\_swift](http://nearthespeedoflight.com/article/2014_09_28_swift)

<http://blog.erratasec.com/2014/06/why-it-had-to-be-swift.html>

# Seriously, why Swift? 2/6

- Objective-C

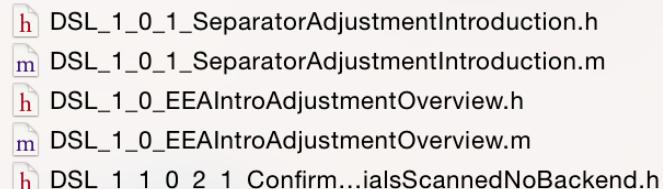
```
NSNumber *myBoolNumber = [NSNumber numberWithBool:YES];  
NSNumber *myBoolNumber = @YES;
```

- So Many Files!
  - at least two files per class [.h | .m]
- Swift
  - one file per class
  - Tuples! :-)

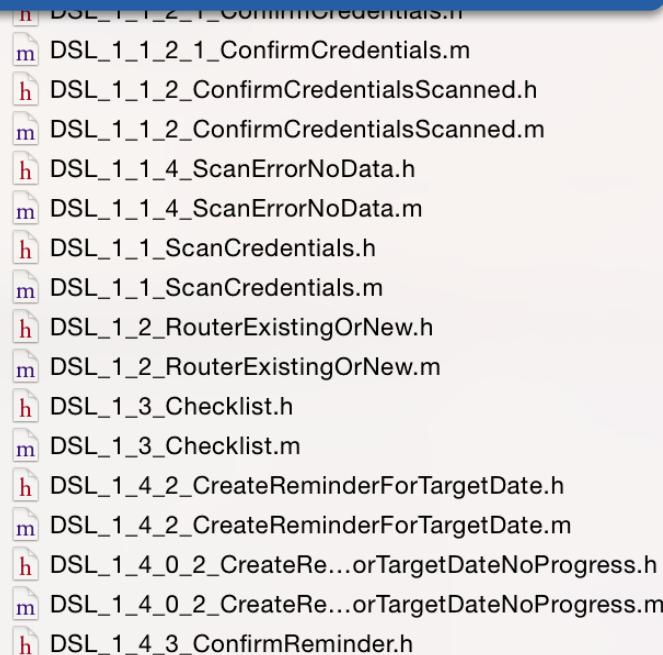
<http://thenewstack.io/some-reasons-why-swift-is-better-than-objective-c/>

[http://nearthespeedoflight.com/article/2014\\_09\\_28\\_swift](http://nearthespeedoflight.com/article/2014_09_28_swift)

<http://blog.erratasec.com/2014/06/why-it-had-to-be-swift.html> 35



DSL\_1\_0\_1\_SeparatorAdjustmentIntroduction.h  
DSL\_1\_0\_1\_SeparatorAdjustmentIntroduction.m  
DSL\_1\_0\_EEAIntroAdjustmentOverview.h  
DSL\_1\_0\_EEAIntroAdjustmentOverview.m  
DSL\_1\_1\_0\_2\_1\_Confirm...ialsScannedNoBackend.h



DSL\_1\_1\_2\_1\_ConfirmCredentials.h  
DSL\_1\_1\_2\_1\_ConfirmCredentials.m  
DSL\_1\_1\_2\_ConfirmCredentialsScanned.h  
DSL\_1\_1\_2\_ConfirmCredentialsScanned.m  
DSL\_1\_1\_4\_ScanErrorNoData.h  
DSL\_1\_1\_4\_ScanErrorNoData.m  
DSL\_1\_1\_ScanCredentials.h  
DSL\_1\_1\_ScanCredentials.m  
DSL\_1\_2\_RouterExistingOrNew.h  
DSL\_1\_2\_RouterExistingOrNew.m  
DSL\_1\_3\_Checklist.h  
DSL\_1\_3\_Checklist.m  
DSL\_1\_4\_2\_CreateReminderForTargetDate.h  
DSL\_1\_4\_2\_CreateReminderForTargetDate.m  
DSL\_1\_4\_0\_2\_CreateRe...orTargetDateNoProgress.h  
DSL\_1\_4\_0\_2\_CreateRe...orTargetDateNoProgress.m  
DSL\_1\_4\_3\_ConfirmReminder.h

# Seriously, why Swift? 3/6

- Objective-C



```
NSString *str = @"hello,";  
str = [str stringByAppendingString:@" world"];
```

- Swift

```
var str = "hello,"  
str += " world"
```

[http://nearthespeedoflight.com/article/2014\\_09\\_28\\_swift](http://nearthespeedoflight.com/article/2014_09_28_swift)

36 <http://blog.erratasec.com/2014/06/why-it-had-to-be-swift.html>

# Seriously, why Swift? 4/6

- Objective-C is dead
- Academic languages suck
- C Syntax is a must
- A stable language
- Cocoa compatible
- Objective-C compatible
- It happened at reference-counting
- Debugging & Playgrounds
- Compile all languages
- At least you got functional



[http://nearthespeedoflight.com/article/2014\\_09\\_28\\_swift](http://nearthespeedoflight.com/article/2014_09_28_swift)

37 <http://blog.erratasec.com/2014/06/why-it-had-to-be-swift.html>

# Seriously, why Swift? 5/6

The syntax of an programming language is about 10- 15%.

Most auf the concepts you see in Swift, you will find also in other languages.

Most of the daily use is fixing some pixel errors and using the Frameworks.

- (void)viewDidLoad {}  
ObjC



override func viewDidLoad() {}  
Swift

# Seriously, why Swift? 6/6

**“Swift is a successor  
to the C and Objective-  
C languages”**

Objective-C without the C.  
“Swift isn't a new language;  
The language it's called Swift  
and it totally rules.”

# Andreas Wittmann

(Swift-Evangelist)

“Any application that can be  
written in Swift will be written in  
**Swift!**”

[@anWittmann](#)

**Swift is the best way  
to write iOS app.**

USTWOGAMES PRESENTS  
**MONUMENT VALLEY**  
 IN NUMBERS

**2,440,076**

OFFICIAL SALES

iOS	1,736,431
Google	296,085
Amazon (paid)	92,247
Amazon (free)	407,560

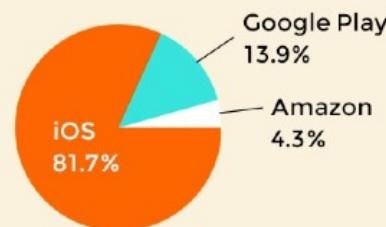
INSTALLED ON OVER  
**10,000,000**  
 UNIQUE DEVICES\*

**575,608**

UPGRADES TO FORGOTTEN SHORES

**\$5,858,625**

REVENUE



HIGHEST ONE-DAY REVENUE

**\$145,530**

APRIL 3, 2014 (LAUNCH DAY)

\* INCLUDES MULTIPLE DEVICES FROM 1 SALE, FAMILY SHARE, UNAUTHORIZED DOWNLOADS, VIA FLURRY

REVENUE OVER TIME

250K

<http://static1.squarespace.com/static/527b69fbe4b0feb0febe4fc9f7/t/54b80483e4b0f32f8253d06c/1421345934518/?format=2500w>

DEVELOPMENT COSTS



ORIGINAL

55 weeks\*

\$852,000



FORGOTTEN SHORES

29 weeks\*

\$549,000

**8** CORE TEAM MEMBERS  
 BASED IN LONDON

\* ACTUAL TEAM SIZE AND HOURS/WEEK VARIED ACROSS THE PROJECT LIFETIMES

**50%**  
 OF PLAYERS WHO  
 STARTED THE GAME  
 COMPLETED IT

LOCALIZED  
 INTO  
**13**  
 LANGUAGES

OVER  
**200**  
 FAN CREATIONS  
 SUBMITTED TO THE  
 MONUMENT FRIENDS  
 TUMBLR

**2,208,418**  
 TOTEMS  
 DROWNED

**24%**  
 OF MV PLAYERS  
 BOUGHT  
 FORGOTTEN SHORES

OVER  
**15,800**  
 STREET FIGHTER MATCHES  
 PLAYED DURING